特别放送：成为DevOps工程师的必备技能（下）

2019-10-19 石雪峰

DevOps实战笔记

进入课程 >

讲述：石雪峰
时长 15:05　大小 13.83M

你好，我是石雪峰。在上一讲，我介绍了 DevOps 工程师的具体职责以及 DevOps 工程师必备的 3 项软实力，分别是沟通能力、同理心和学习能力。有了这些认知之后，我们今天来看看“重头戏”：DevOps 工程师必备的硬实力以及学习路径。

DevOps 工程师必备的硬实力

所谓硬实力，说白了就是指一个人的技术能力。软实力通常是“只可意会不可言传”的，但技术本身就具体多了，重要的是，技术水平的高低相对来说也更好衡量。在公司里面，技术人员要想获得晋升，重点就是依靠技术能力。

IT 行业覆盖的技术领域非常广，而且近些年的新技术也是层出不穷的，从入门到精通任何一门技术，都需要大量时间和精力的投入。那么，在面对这么多技术的时候，究竟要选择从

哪个开始入手，真是一个难题。对于希望成为 DevOps 工程师，甚至是 DevOps 专家的你来说，究竟有哪些必须掌握的核心技术呢？

1. 代码能力

现在这个时代，代码能力可以说是最重要的硬实力了。IT 行业自然不用说，像运维有运维开发，测试也有测试开发，就连产品经理都要懂代码，不然可能都没办法跟开发同学顺畅交流。

对于工具平台自身的建设而言，代码能力自然是重中之重。这不仅仅在于通过写代码来实现工具平台本身，还在于你能了解开发的完整过程。这些平台的用户每天跟代码打交道的时间可能比跟人打交道的时间还多，如果你不能理解他们的日常工作方式，那么你做出来的工具平台，又怎么能真正解决团队的问题呢？

这里提到的**代码能力包含两个方面，分别是脚本语言能力和高级语言编程能力。**

> **脚本语言能力。**这对于运维工程师来说自然是驾轻就熟，各种 VIM、Emacs 手到擒来，Shell 和 Python 也是轻车熟路。而对于开发人员来说，难点不在于语法本身，而在于对关联操作系统和命令的理解上。毕竟，脚本语言是一种快速的自动化手段，追求的是高效开发，简单易用。
>
> **高级语言编程能力。**你需要至少掌握一门高级语言，无论是 Java、Python 还是 Ruby 和 PHP。**其实语言只是工具，你不用过度纠结于选择哪门语言，要求只有一个，就是你能用它来解决实际问题**，比如能够支持你实现面向移动端或者 Web 端的工具平台开发。**为了写出好代码，而不仅仅是写出能用的代码，你也需要对于一些常见的开发框架和开发模式有所了解。**这是一个相对漫长的过程，绝对不是什么“21 天精通 XX 语言”就够了。因为看得懂和写得好，完全是两码事。

好的代码是需要不断打磨和推敲的。**与其说写好代码是一门技术，不如说是一种信仰。**我们团队的内部沟通群名叫作“WBC 团队”，“WBC”也就是“Write Better Code”的缩写，这其实也是我们团队对自己的一种激励。在日常的开发过程中，我们会不断发现和总结更好的实现方式，在内部分享，互相学习，从而持续提升代码能力。我截取了一部分我们最近优化流水线脚本的经验总结，你可以参考一下。其实，每个人都能总结出自己的代码心经。

优化流水线脚本的经验总结
1.封装方法拆分到最小粒度，以每个小功能方法组合业务逻辑。
2.尽量少使用def关键字，多使用对象类型或基本类型定义变量。
3. 尽量使用String类型赋值原子参数。
4. 尽量使用Groovy封装操作方法，减少脚本依赖调用。
5. 多利用监听方法或公共方法来扩展你的能力。
6. 所有Pipeline Script的方法要在抽象类中增加封装方法。
7. 编码从上往下设计，从下向上实现。
8. 实现新方法时，要先找从已有的方法中寻找。

2. 自动化能力

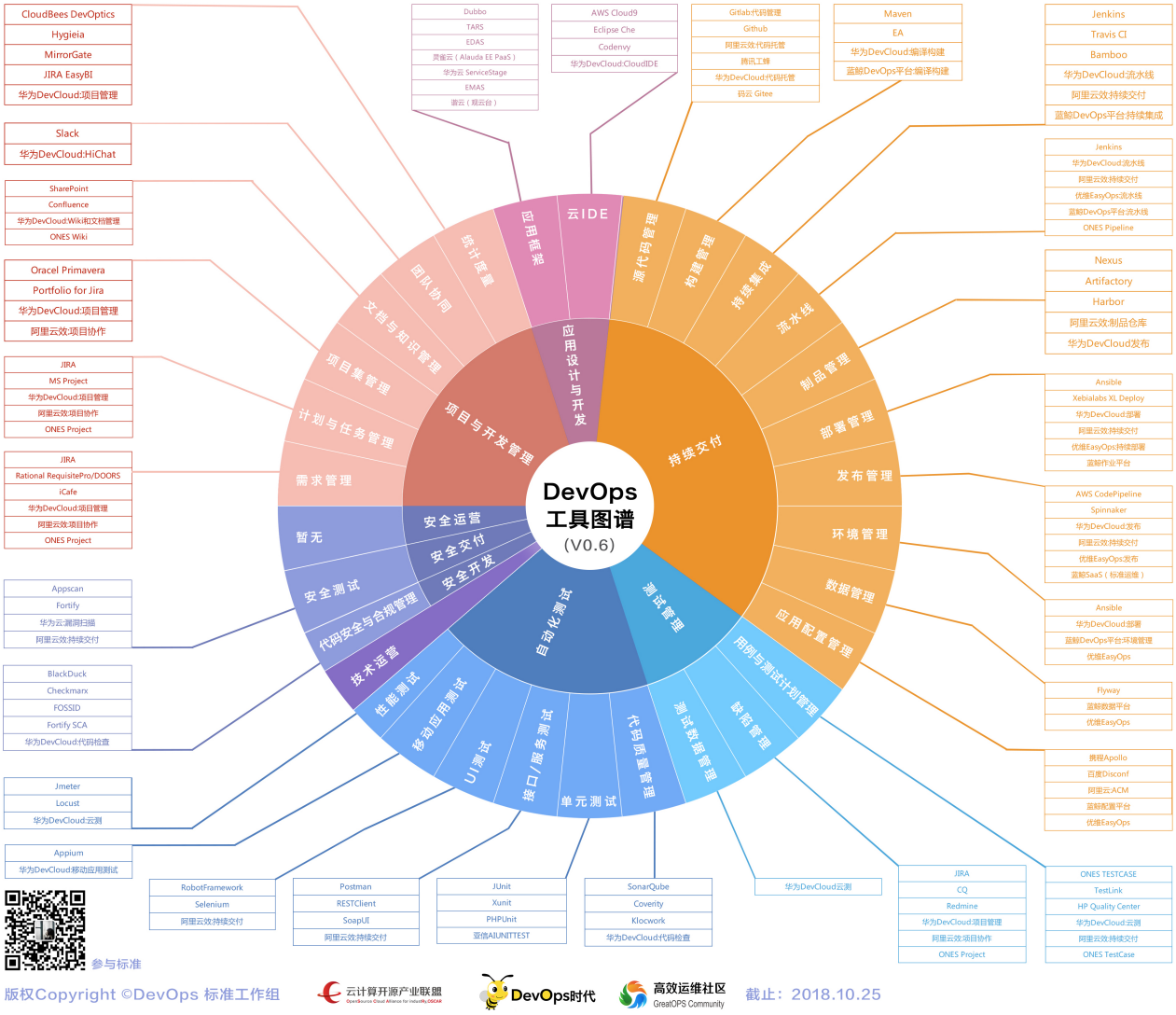
在自动化方面，你首先需要对 CI/CD，也就是持续集成和持续交付，建立起比较全面的认知。因为 CI/CD 可以说是 DevOps 工程领域的核心实践，目前大部分公司都在集中建设软件的持续交付能力，尤其是以流水线为代表的持续交付平台，很多时候就同 DevOps 平台划上了等号。

接下来，为了实现全流程的自动化，你需要能够熟练使用 CI/CD 各个关键节点上的典型工具，并且了解它们的设计思路。

一方面，目前很多公司都在拥抱开源，参与开源，开源工具自身的成熟度也非常高，并且逐渐取代商业工具，成为了主流方案。通过直接使用开源工具，或者基于开源工具进行二次开发，也是自动化领域投入产出比最高的方式。所以，像版本控制工具 Git、代码托管平台 Gitlab、CI 工具 Jenkins、代码扫描工具 Sonar、自动化配置管理 Ansible、容器领域的 Docker、K8S 等等，这些高频使用的工具都是你优先学习的目标。

另一方面，**无论是开源工具，还是自研工具，工具与工具之间的链路打通也是自动化的重要因素。**所以，在理解开源工具的实现方式的基础上，就要能做到进可攻，退可守。无论是封装，还是自研，有了工具的加持，CI/CD 也会更加游刃有余。

关于 DevOps 的工具图谱，我跟你分享一个信通院的 DevOps 能力成熟度模型版本，供你参考。值得注意的是，**工具不在多，而在精。**其实，工具的设计思路和理念有共通之处，只要精通单个节点上的工具，就可以做到以点带面。



3.IT 基础能力

我始终认为，运维是个特别值得尊敬的工种，也是 DevOps 诞生的原点。如果你不是运维出身，那你要重点掌握运维的基础概念，最起码要了解**Linux 操作系统方面的基础知识**，包括一些**常用的系统命令使用**，以及**网络基础和路由协议**等。毕竟，对于开发者来说，他们通常习惯基于 IDE（集成开发环境）图形界面工作。比如，如果问一个 iOS 开发同学怎么通过命令行的方式进行构建调试，或者如何用代码的方式实现工程的自动化配置，他可能就答不上来了。

另外，随着基础设施即代码的技术不断成熟，你还要能看懂环境的配置信息，应用自动化构建、运行和部署的方式等，甚至可以自行修改环境和应用配置，这样才能实现所谓的开发自运维。虽然在大多数公司，运维的专业能力一般都会通过运维平台对外提供服务，但对于基础概念，还是需要既知其然，也知其所以然。

4. 容器云能力

云计算对于软件开发和部署所带来的变化是革命性的。未来企业上云，或者基于云平台的软件开发会慢慢成为主流。而容器技术又天生适合 DevOps，Kubernetes 可以说是云时代的 Linux，基于它所建立的一整套生态环境，为应用云化带来了极大的便利。

所以，无论是容器技术的代表 Docker，还是实际上的容器编排标准 Kubernetes，你也同样需要熟悉和掌握。尤其是在云时代，基于容器技术的应用开发和部署方式，都是 DevOps 工程师必须了解的。

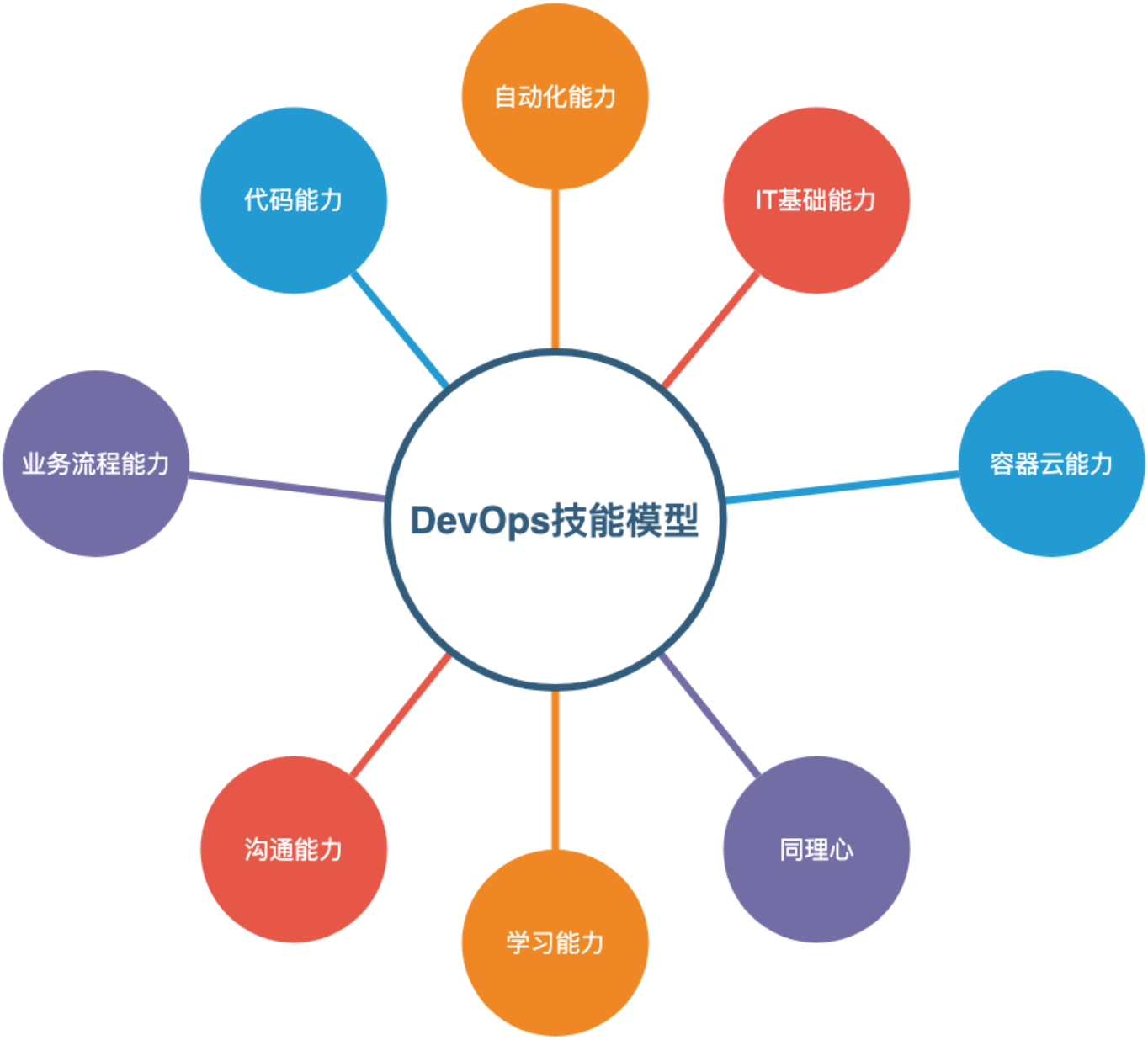
5. 业务和流程能力

在任何时候，DevOps 的目标都是服务于业务目标，DevOps 本身也从来不是墨守成规的方式，而是代表了一种变革的力量。所以，加强对业务的理解，有助于识别出 DevOps 改进的重点方向，而流程化的思维建设，有助于突破单点，放眼全局。

很多时候，**企业需要的不仅仅是一个工具，而是工具所关联的一整套解决方案，其中最重要的就是业务流程。**

对于 DevOps 工程师来说，**要有能力发现当前流程中的瓶颈点，并且知道一个更加优化的流程应该是怎样的**，这一点也是制约工程师进一步拓展能力的瓶颈之一。

举个例子，对于开发 DevOps 平台工具来说，你可能认为最合适承担的团队就是开发团队，因为他们的代码能力最强。但是实际上，DevOps 平台的设计，很多时候都是由最熟悉企业内部研发流程的团队来主导的。正因为 DevOps 工程师的工作应该同业务紧密联系，更加关注于全局交付视角，所以很多时候，配置管理、质量管理、项目管理和技术运维团队更多地在承担相近的角色。**毕竟，只有方向正确，所做的一切才是加法。**



学习路径

那么，要想成为 DevOps 工程师，是否有一条普适性的学习路径呢？实际上，这个问题就跟我们要在公司推行 DevOps，是否存在一条通用的改进路径一样，并不是一个容易回答的问题。

从前面的能力模型可以看出，DevOps 工程师特别符合现在这个时代的要求，他具备多重复合能力，是典型的全栈工程师，或者“梳子型”人才。因为只有这样，才能充分弥合不同角色之间的认知鸿沟，堪称团队内部的万金油。

基于过往在公司内部推行 DevOps 的经验，以及当前行业的发展趋势，我有几条建议送给你：

1. 集中强化代码能力

未来的世界是软件驱动的世界。我们以前总说的必备能力，比如外语、开车等，未来都可以被软件所取代。而编程能力即将成为下一个必备能力，甚至连国务院发布的《新一代人工智能发展规划》中都提到，要在中小学普及推广编程教育。

而写可以用的代码，和写好的代码之间，距离绝不只是一点点而已。你可能会说，以后都用人工智能来编程了，可问题是人工智能从何而来？又是谁来训练和标注人工智能的呢？所以，越是基础的能力，越不会过时，比如数学、核心的编程思想、数据结构，以及基于代码构建对世界的认知和建模能力。

所以，如果你现在只是刚开始接触代码，我建议你给自己定一个目标，专门强化自己的代码能力，至少花 1 年时间，从新手变成熟手，这对于你未来在 IT 行业的发展，至关重要。

跟你分享一个小技巧。你可以基于成熟的开源软件来边学习边应用，比如像 Adminset 这种轻量级的自动化运维平台，已经可以解决大多数中小公司的问题了。其实，代码能力不仅仅是掌握语法和框架，更重要的是基于场景，整体设计数据和业务流程，并通过代码实现出来。毕竟，只有结合实际的应用场景进行学习，才是最有效率的。

2. 培养跨职能领域核心能力

相信经过几年的工作，你已经具备了当前岗位所需要的基本能力，这是你当前赖以为生的根本。那么在这些能力的基础上，逐步发展跨领域跨界的能力，尤其是那些核心能力，就成了投入产出比最高的事情。

举个例子，如果你是软件开发工程师，那么恭喜你，你已经走在了代码的道路上，接下来，运维能力就是你要尝试攻克的下一个目标。而在这些目标中，比如操作系统、自动化部署以及云能力，就是你要最优先发展的跨界能力，因为它们是运维的核心，也是了解运维最好的出发点。反过来说，如果你从事的是运维行业，那么除了常用的脚本以外，核心代码能力就是你的目标。

其实，我们每天的工作其实都离不开跨界，比如，运维每天部署的应用，为什么要部署这么多实例？每个实例之间的调用关系又是怎样的？**多问几个为什么，往往就有新的收获。**

不仅如此，在接触跨领域的时候，除了基础核心技能，那些最常见的工具，你也要花时间来了解。现在网上的资料足够多，快速入门应该并不困难。

3.DevOps 核心理念和业务思维

如果你不理解 DevOps 到底是什么，那何谈成为 DevOps 工程师呢？因此，像 DevOps 中的核心理念，比如精益敏捷、持续交付，以及很多实践，你都要有所了解。当然，如果你订阅了这个专栏，我将带你走过前面的这段路，你可以快速地进入下一阶段，在实战中练习。

DevOps 在公司的落地是大势所趋，也许你所在的团队也会参与其中，那么除了做好自己的本职工作外，你也可以多参与，多思考，看看推进的过程是怎样的，涉及到的角色又在做些什么，项目的整体进展和计划是什么。在实战中练习和补齐短板，对于积累经验来说，是不可或缺的。**很多时候，不是没有学习的机会，只是我们自己不想看到罢了。**

另外，可能你现在距离业务还比较远，那么你可以尝试了解一些大的业务目标，多跟你所在团队的上下游进行沟通，看看他们现在的关注点在什么地方。既然业务的目标需要整个团队紧密协作才能完成，那么每个团队都是其中的一份子，所以他们身上也同样体现了业务的目标。

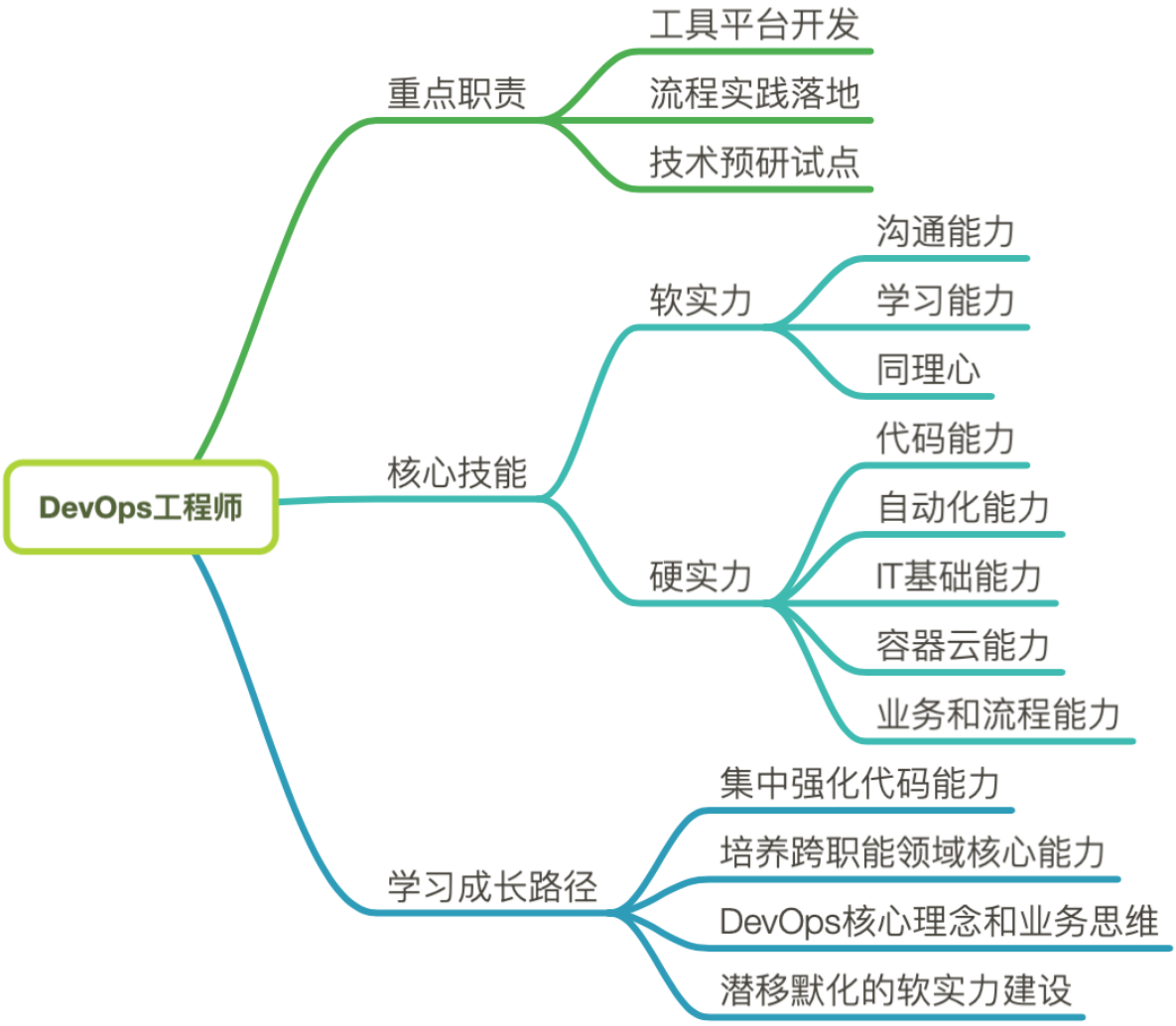
4. 潜移默化的软实力建设

类似沟通能力、同理心、自驱力、学习能力、主动性等，无论从事任何职业，都是你身上的闪光点。很多天生或者从小养成的习惯，需要长时间潜移默化的训练才能有效果。

很多时候，IT 从业人员给人的印象都是不善表达，再加上东方文化的影响，本身就比较含蓄，这对很多沟通和表达来说，都是潜在的障碍。这个时候，就要尽量把握已有的机会，比如多参加团队内部的读书分享、公司内部的讲师培训报名等。即便刚开始分享的内容还不足你脑中的 1%，但至少也是一个好的起点。我的建议就是 6 个字：**勤练习，多总结**。就像 DevOps 一样，持续改进和持续反馈，培养自己的自信心。

总结

总结一下，我在这两讲给你介绍了 DevOps 工程师要重点关注的 3 大职责，分别是工具平台开发、流程实践落地和技术预研试点。另外，我还基于实用角度提炼了 8 大核心能力模型，分为 3 条软实力和 5 条实力，并给出了 4 条提升 DevOps 核心能力的建议。为了方便你复习和理解，我画了一张脑图，把这两讲内容进行了汇总，你可以参考一下。



最后，我想强调的是，就像 DevOps 没有明确的定义一样，DevOps 工程师的技能也没有明确的限定，所以，你要时刻保持好奇心，持续学习，总结出自己的能力体系，并在实践积累经验，这样才能在激烈的竞争中占得先机。

思考题

针对我们这两讲的内容，你觉得自己需要提升哪方面的能力呢？你有哪些快速提升能力的小窍门吗？

欢迎在留言区写下你的思考和答案，我们一起讨论，共同学习进步。如果你觉得这篇文章对你有所帮助，欢迎你把文章分享给你的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　特别放送：成为DevOps工程师的必备技能（上）

精选留言 (6)

写留言

Robert小七
2019-10-19

老师分享的devops工具图谱大部分都是一些商业化的平台中的模块，有没有专门针对开源的工具图谱？我觉得如果提供开源工具图对订阅专栏的同学更好！

展开

作者回复: 开源工具的图谱网上应该有很多，我也不是每个工具都使用过，所以建议还是贵精不贵多，可以先把几个环节的典型工具跑通并真正运用起来，再根据需要评估新的工具，如果你对具体哪个方面，或者解决那些问题的工具感兴趣，欢迎给我留言，看看能否给你一些有针对性的推荐哈！

1

Snake潜
2019-10-19

想问一下后面会讲具体的实施么？比如CI用什么如何整合？还有希望有讲一下，比如亚马逊是如何做CICD的。用到什么技术？怎样保证持续开发的稳定性。谢谢啦

展开

作者回复: 你好，后面会有专门一讲来介绍CI持续集成的内容，其实从工具层面来说，现在的解决方案都比较成熟，通过关联版本控制系统和持续集成系统就可以实现每次集成的自动化任务。在这个过程中，有几个点需要特别关注：

1. 测试能力的集成，如果CI环境中缺乏测试能力，那么就无法做到持续开发的稳定性
2. 质量门禁的集成，如果没有设定质量门禁，那么持续集成中发现的问题就不会在第一时间解决
3. 通知机制的集成，需要建立一种有效的通知机制，比如即时通讯软件来告知CI结果

当然，工具，集成这些方面的因素，更重要的是培养团队的习惯，这也是落地CI最大的挑战。

至于亚马逊的CICD实践，我这边了解不多，但核心几个方面，强大的内部工具支持，应用架构充分解耦，全功能团队和研发自运维。

1

玉剑冰锋

2019-10-21

从认知到落地相信这一定是个漫长的过程，作为个人来讲，在公司都没开始推行devops或者都还没意识到的时候，日常工作学习中应该如何去实践，比如CI/CD,只是自行搭建环境测试学习跟线上还是有很大区别

鲍建飞

2019-10-20

你好，我之前一直有一个疑问，怎样能快速理解docker和k8s之间的关系呢？看了很久也没把认知提上去，那个编排还是很难理解

展开

大鼻子先生

2019-10-19

老师，操作系统和网络协议以及jenkins方面有相关的书籍推介吗

作者回复: 你好，Jenkins方面的书籍说实话一直比较少，我之前翻译过一本Jenkins2权威指南，应该是目前业界比较权威的书籍了，你可以看下。关于操作系统和网络协议，如果你不是要求特别精神的话，我个人认为单纯了解的层面，鸟哥的书就足够啦。

Robert小七

2019-10-19

老师后面讲模块时，会讲京东云是怎么做的吗？对京东云有过一些基本的了解，很想知道京东云面对配置管理是如何做的！devops讲一切相关的元数据都纳入版本控制，但是实际落地时很难！

> 作者回复: 你好，你觉得将元数据纳入版本控制的难点具体在什么地方呢，我刚好在这方面有很多积累，可以提出你的问题，我帮你一起分析一下哈。